

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:
Evan P. Ireland

Serial No.: 10/709,917

Filed: June 4, 2004

For: Attribute-Based Component
Programming System and Methodology for
Object-Oriented Languages

Examiner: Wang, Rongfa Philip

Art Unit: 2191

REPLY BRIEF

Mail Stop Appeal
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

REPLY BRIEF ON BEHALF OF EVAN P. IRELAND

This is an appeal from the Final Rejection mailed September 12, 2008, in which currently pending claims 1-61 stand finally rejected. Appellant filed a Notice of Appeal on December 12, 2008 and an Appeal Brief on February 10, 2009. This Reply Brief is submitted electronically in response to the Examiner Answer mailed on April 28, 2009.

TABLE OF CONTENTS

1. STATUS OF CLAIMS	3
2. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL	3
3. ARGUMENT IN REPLY TO ANSWER.....	3
A. First Ground: Claims 1-46 rejected under 35 U.S.C. 112, second paragraph	3
B. Second Ground: Claims 1-5, 15-19, 21-26, 28-30, 33, 40-43, and 45-46 rejected under 35 U.S.C. 103(a)	3
C. Third Ground: Claims 6-14, 20, 31-39, and 44 rejected under 35 U.S.C. 103(a)	6
D. Fourth Ground: Claims 47-61 rejected under 35 U.S.C. 103(a)	8
E. Conclusion	9

1. STATUS OF CLAIMS

Claims 1-61 are pending in the subject application and are the subject of this appeal. An appendix setting forth the claims involved in the appeal is included as Section 8 of Appellant's previously filed Appeal Brief.

2. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to be reviewed on appeal are stated in Appellant's Appeal Brief filed on February 10, 2009; provided, however, that the Examiner has now withdrawn the 35 U.S.C. Section 112, second paragraph rejection of Claims 1-46.

3. ARGUMENT IN REPLY TO ANSWER

A. First Ground: Claims 1-46 rejected under 35 U.S.C. 112, second paragraph

The Examiner withdrew the rejection of Claims 1-46 under 35 U.S.C. Section 112, second paragraph after the filing of Appellant's Appeal Brief.

B. Second Ground: Claims 1-5, 15-19, 21-26, 28-30, 33, 40-43, and 45-46 rejected under 35 U.S.C. 103(a)

In the Answer, the Examiner continues to argue that DeGroot's declarative statements are comparable to the attributes of Appellant's attribute-based component programming solution. The Examiner contends that DeGroot's "declarative statements" comprise "information" that can be used to specify declaratively behavior to be added to a program. However, the Examiner's argument ignores the fact that in response to the Examiner's prior Office Action in this case, Appellant specifically amended the claims in an effort to avoid any such interpretation by specifying that the "attributes" of Appellant's claimed invention comprise active metadata that is used to dynamically generate program code for inclusion in a program. For instance, Appellant's claim 1 includes the following claim limitations:

A method for dynamically generating program code adding behavior to a program based on attributes, the method comprising:

adding a static field of type Component to a program class of the program to create a component;

defining at least one attribute specifying declaratively behavior to be added to the program, wherein said at least one attribute comprises active metadata used to generate program code for inclusion in the program;

associating said at least one attribute with the component; and

in response to instantiation of the component at runtime, generating a subclass based on the program class and said at least one attribute, the subclass including dynamically generated program code based on said at least one attribute

(Appellant's claim 1, emphasis added)

As shown above, the attributes of Appellant's invention comprise active metadata that is used to dynamically generate program code for inclusion in a program that adds behavior to the program (also see e.g., Appellant's specification, paragraph [0085]). In the Answer, the Examiner argues that DeGroot includes these same teachings by stating:

Per specification, [0126], line 4, "attribute can be considered as active metadata". For at least the reasons above, DeGroot discloses attribute and therefore active metadata. In claim 1, for example, claims "active metadata is used to generate program code". DeGroot's declarative statements are used to augment functionality of objects and therefore are considered metadata used to generate program code.

(Examiner Answer, paragraph B2, page 29, emphasis added)

By pointing to the specification [0126], line 4, the Examiner is attempting to rely on Appellant's own specification to provide teachings not found in DeGroot – namely that an attribute comprises active metadata used to dynamically generate program code. However, it is well settled that the Examiner cannot rely upon a patent applicant's disclosure in reaching an obviousness determination under 35 U.S.C. Section 103 (see e.g., MPEP Section 2142). Moreover, in this case the Examiner is not only using impermissible hindsight, the Examiner goes even further by, in essence, referencing Appellant's specification for teachings that are not actually found in the prior art references. In particular, Appellant's attributes are described as "active" metadata that are used as the basis for dynamically generating program code. As discussed in detail in

Appellant's Appeal Brief, DeGroot does not teach dynamically generating program code. Instead, with DeGroot's system a developer/user must write and submit one or more declarative statements to augment the functionality of a given method (see e.g., DeGroot, col. 3, line 64 to col. 4, line 5 and col. 6, lines 24-25 (rules expressed using basic programming language)).

Additionally, Appellant's invention also specifically provides for dynamically generating a subclass that includes dynamically generated program code based on the defined attribute (see e.g., Appellant's claim I, quoted above). The Examiner acknowledged that DeGroot does not include these teachings and, therefore, the Examiner added McGurrin as providing these teachings. However, in the Answer the Examiner does not traverse Appellant's argument that McGurrin does not, in fact, provide these teachings. Instead, the Examiner simply asserts that DeGroot discloses dynamically generating program code and, therefore, the combination of DeGroot and McGurrin discloses dynamically generating program code based on defined attributes (Examiner Answer, paragraph B4 - Examiner Response, page 30). Respectfully, the Examiner cannot have it both ways. The Examiner cannot acknowledge at page 4 of the Answer that DeGroot does not include all of the cited limitations of Appellant's claims and then rely on DeGroot as teaching all of the limitations at page 30 of the Answer. As the Examiner has not traversed Appellant's argument regarding McGurrin, Appellant believes that the Examiner now recognizes that McGurrin does not cure the deficiencies of DeGroot, as McGurrin fails to teach or suggest generating a subclass that includes dynamically generated program code based on the defined attribute.

In summary, neither DeGroot nor McGurrin, taken singly or in combination, include any teaching or suggestion of defining attributes that comprise active metadata and using these attributes for dynamically generating subclasses including code for adding behavior to a program as it executes at runtime. Therefore, it is again respectfully submitted that claims 1-5, 15-19, 21-26, 28-30, 33, 40-43, and 45-46 distinguish over the cited references and the Section 103 rejection should not be sustained.

C. Third Ground: Claims 6-14, 20, 31-39, and 44 rejected under 35 U.S.C. 103(a)

1. Claims 6, 10, 20, 31, 35 and 44

As to the claims included in Appellant's Third Ground of Appeal, the Examiner argues in the Answer that there is no language including an attribute "class" being generated in claim 6 and therefore such argument is outside the scope of the claim language. Appellant acknowledges that the claim limitations of claim 6 (and similar claim 31) do not specifically include limitations of a "class" (although other claims such as 7-9, 11-14, 32-34 and 36-39 do include such limitations as discussed below); however, this is not the central point of Appellant's argument. Instead, the primary thrust of Appellant's argument is that the attributes described by Foster are dramatically different than those of Appellant's claimed invention (discussed in detail above and in the Appeal Brief with respect to Appellant's **Second Ground** of Appeal). In particular, Foster provides no teaching or suggestion of attributes included as comments in source code (or in an external properties file), which are used as the basis for generating code for adding behavior to a software program as it executes at runtime. In this regard, Appellant notes that claims 6 and 31 are dependent claims that must be read in conjunction with independent claims 1 and 26, which provide that the attributes comprise active metadata for dynamically generating program code. Foster includes no comparable teaching of generating program code based on attributes defined in source code comments (or an external file) as discussed in detail in Appellant's previously-filed Appeal Brief. Additionally, Foster's attributes are **not included as comments in program source code**, but rather are included as tags in a version management file.

2. Claims 7-9, 11-14, 32-34 and 36-39

Claims 7-9, 11-14, 32-34 and 36-39, specifically include claim limitations of a class containing attributes derived from source code comments (or an external properties file). Accordingly, the Examiner's position in the Answer that Appellant's argument is not supported by the claims cannot be applied to these claims, which specifically include such limitations. For example, Appellant's claim 7 includes claim limitations of precompiling a class containing static attributes from comments, Appellant's claim 8 further provides that the class containing static attributes is loaded before subclass

generation, and claim 9 adds limitations of an automated mapping between attribute syntax in comments and attribute syntax as expressed in generated program code. Claims 32-34 include similar limitations, while claims 11-14 and 36-39 are similar but provide that the attributes are defined in a property file external to the application.

Additionally, the Examiner relies on DeGroot as including teachings of precompiling a class containing attributes from comments. However, given that the Examiner acknowledges that DeGroot does not teach defining attributes in source code comments, it seems apparent that DeGroot cannot teach precompiling attributes from comments. Additionally, the teachings of DeGroot referenced by the Examiner as including teachings comparable to Appellant's claim 7 (see Answer, paragraph 2, page 15) do not discuss precompiling a class containing attributes. Instead, DeGroot describes that as subclassing technique requires "recompiling" the code (which in turn requires shutting down the application, reloading the code, etc.) a better approach to adding behavior is desired (DeGroot, col. 2, lines 40-46). Thus, Appellant respectfully believes the teachings of DeGroot are not at all comparable. Further, by relying on DeGroot, the Examiner presumably acknowledges that the added Foster reference does not teach or suggest such features.

The Examiner also argues that DeGroot includes teachings of loading a class containing static attributes before subclass generation as provided, for example, in limitations of Appellant's claim 8. However, DeGroot in fact teaches away from Appellant's claimed invention by rejecting the notion of subclassing technique as undesirable (DeGroot, col. 2, lines 40-46). With respect to Appellant's claim limitations of an automated mapping between attribute syntax in comments and attribute syntax as expressed in generated program code, the Examiner references DeGroot's teachings of "maps" (see e.g., Examiner Answer, paragraph 2, page 15). However the referenced teachings of DeGroot is the brief description of Fig. 9 indicating that DeGroot's Fig. 9 is a block diagram "illustrating a plurality of bit maps...". (DeGroot, col. 3, lines 43-45). Respectfully, these teachings of "a plurality of bit maps" have nothing whatsoever to do with the features included as limitations of Appellant's claim 9 other than the fact that the word "maps" from Appellant's claim 9 may be found in the referenced text when one

performs a key word search.

3. Conclusion.

Therefore, for the reasons set forth above, Appellant respectfully submits that the claims distinguish over the combined references and the rejection under Section 103 should not be sustained.

D. Fourth Ground: Claims 47-61 rejected under 35 U.S.C. 103(a)

In the Final Rejection, the Examiner added Zhang as purportedly teaching the following limitations of Appellant's claim 47 which the Examiner acknowledges are not taught by DeGroot and McGurrin:

storing said at least one attribute in a properties file external to the application;
creating a dynamic attributes class based on the properties file;
compiling the application and the dynamic attributes class

(Examiner Answer, paragraph 3, page 22)

However, the Examiner does not traverse Appellant's detailed discussion of Zhang in the Appeal Brief. Instead, the Examiner simply refers to paragraph B2 of the Examiner's response (discussed in the **Second Ground** of Appeal above in this Reply Brief and in Appellant's previously filed Appeal Brief) and states that DeGroot discloses active metadata for augmentation of program code (Examiner Answer, paragraph D1 – Examiner response, page 32). Respectfully, the Examiner cannot both rely on DeGroot as including the teachings of claims 47-61 while at the same time acknowledging that DeGroot does not teach all of the limitations of Appellant's claims. As the Examiner has acknowledged that DeGroot and McGurrin do not include teachings corresponding to the limitations of Appellant's claims 47-61 and has not traversed Appellant's argument that Zhang also fails to include such teachings, Appellant respectfully submits that the rejection under Section 103 should not be sustained.

E. Conclusion

In view of the above remarks and those set forth in the Appeal Brief previously filed by Appellant in this case, it is respectfully submitted that the Examiner's rejection of Appellant's claims under 35 U.S.C. Section 103 should not be sustained. If needed, Appellant's undersigned attorney can be reached at (925) 465-0361.

Respectfully submitted,

Date: June 18, 2009

/G. Mack Riddle/

G. Mack Riddle; Reg. No. 55,572
Attorney of Record

925 465 0361
925-465-8143 FAX